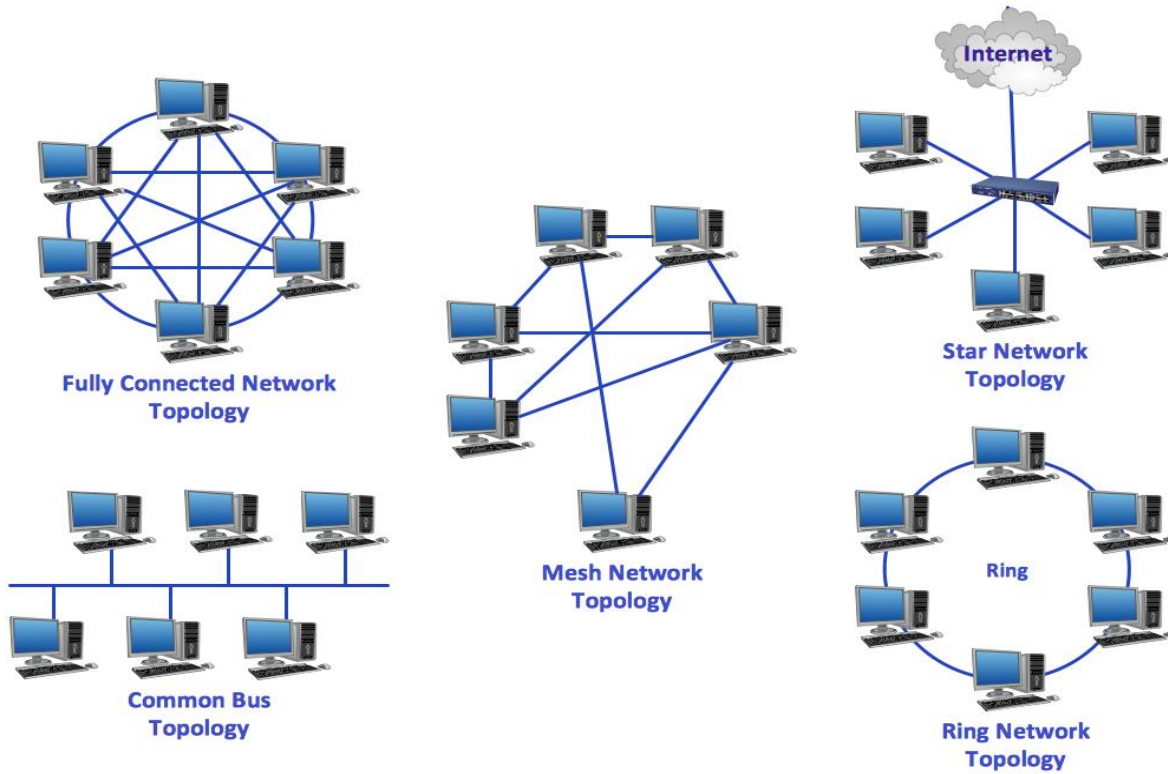


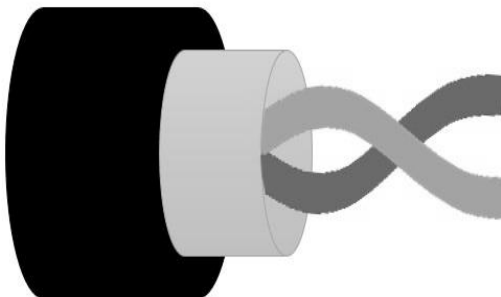
# CN Diagrams

## Module-1

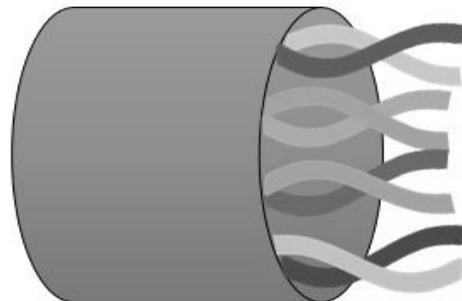


## Module-2

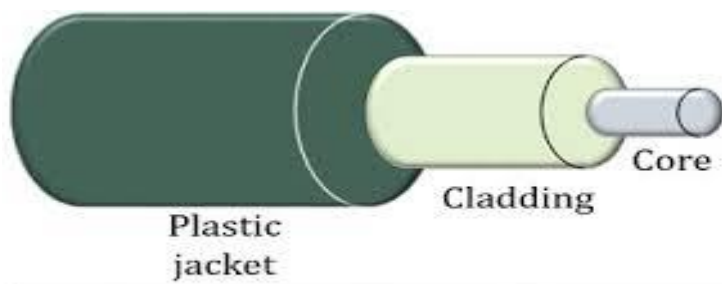
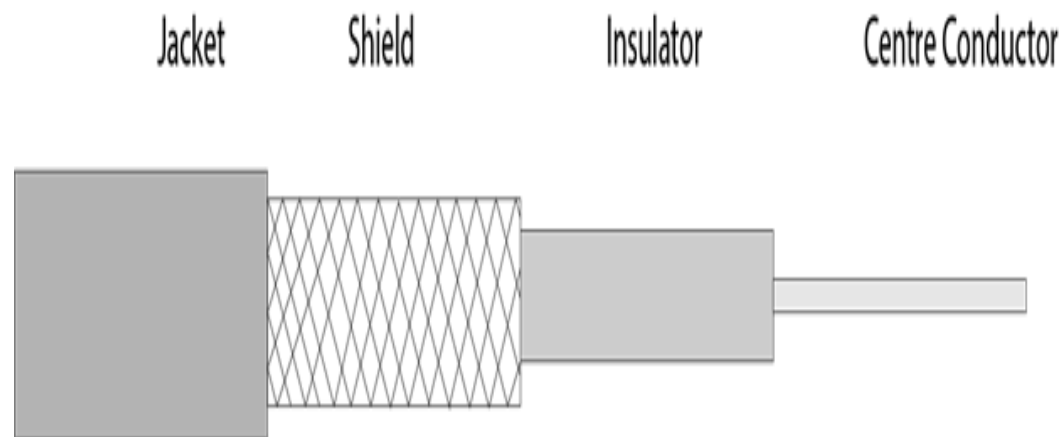
Shielded Twisted Pair Cable



Unshielded Twisted Pair Cable

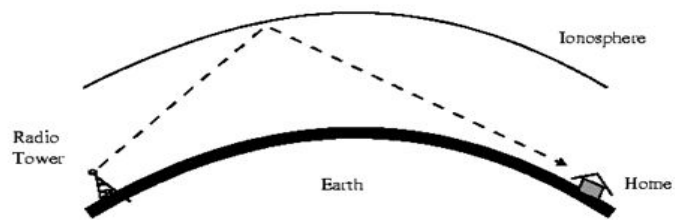


## Coaxial cable

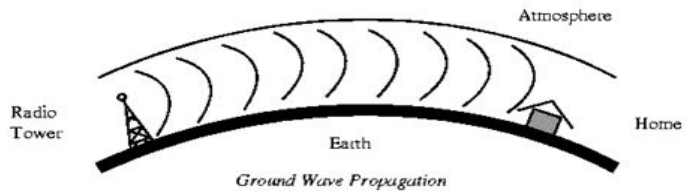


Structure of Optical Fiber

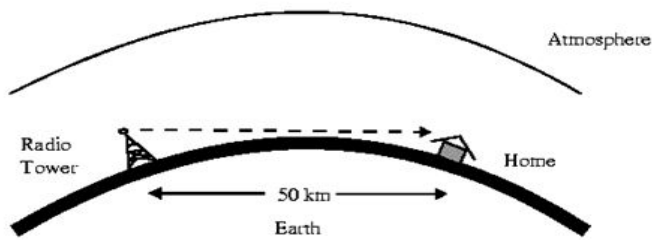
Circuit Globe



skywave

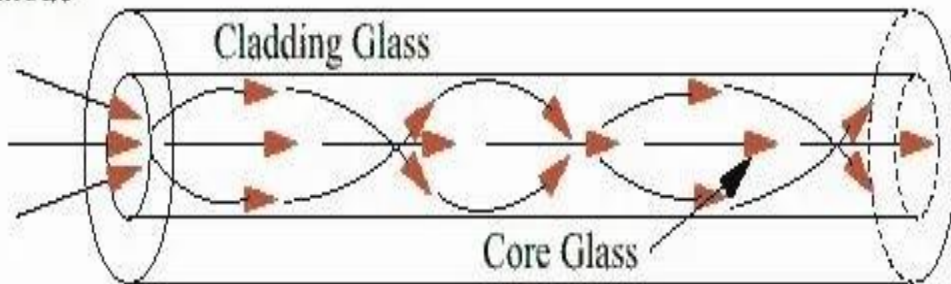


groundwave

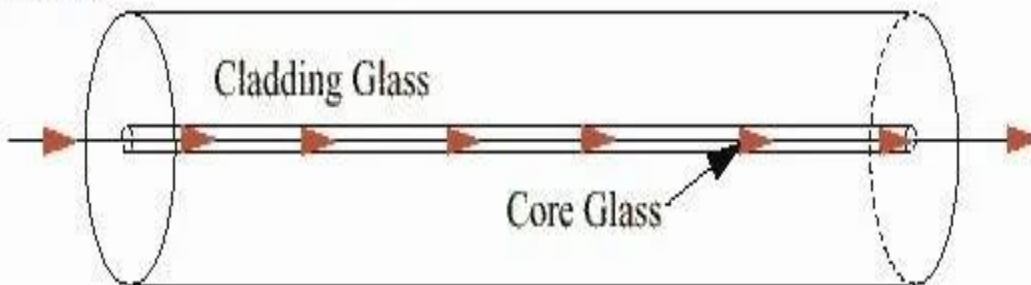


line-of-sight

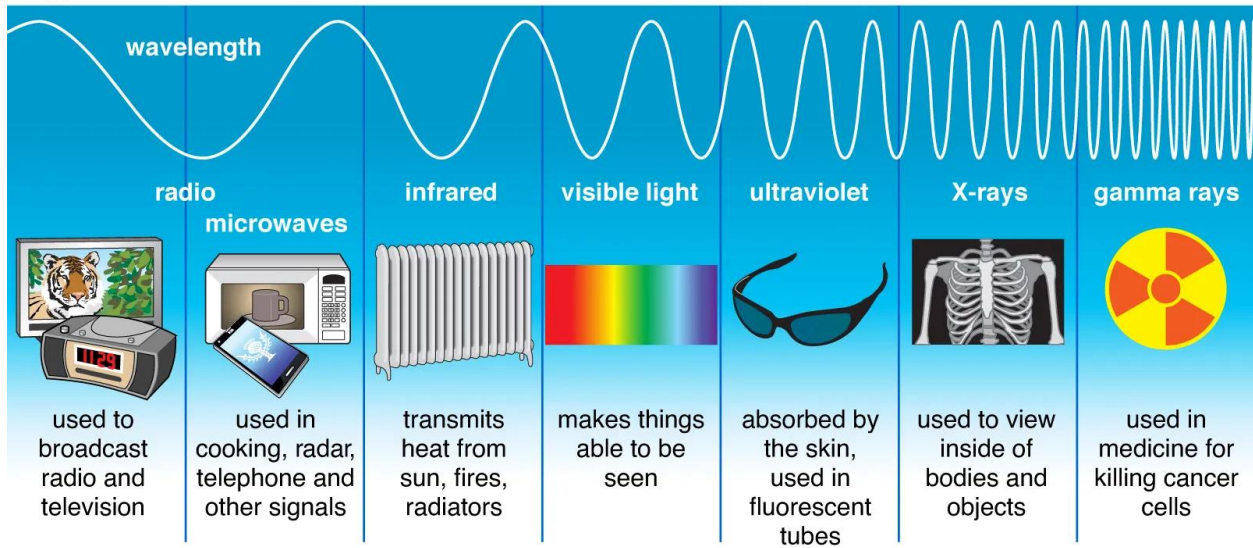
### Multimode



### Single-Mode



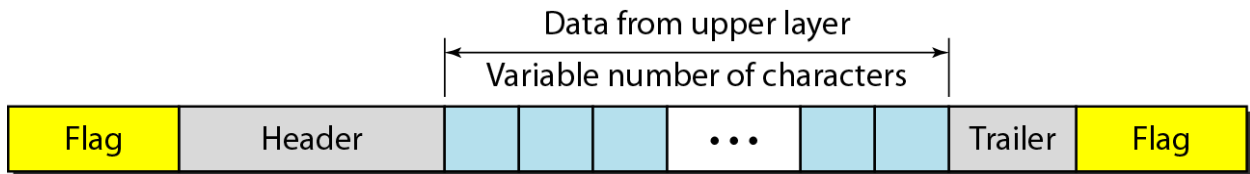
## Types of Electromagnetic Radiation



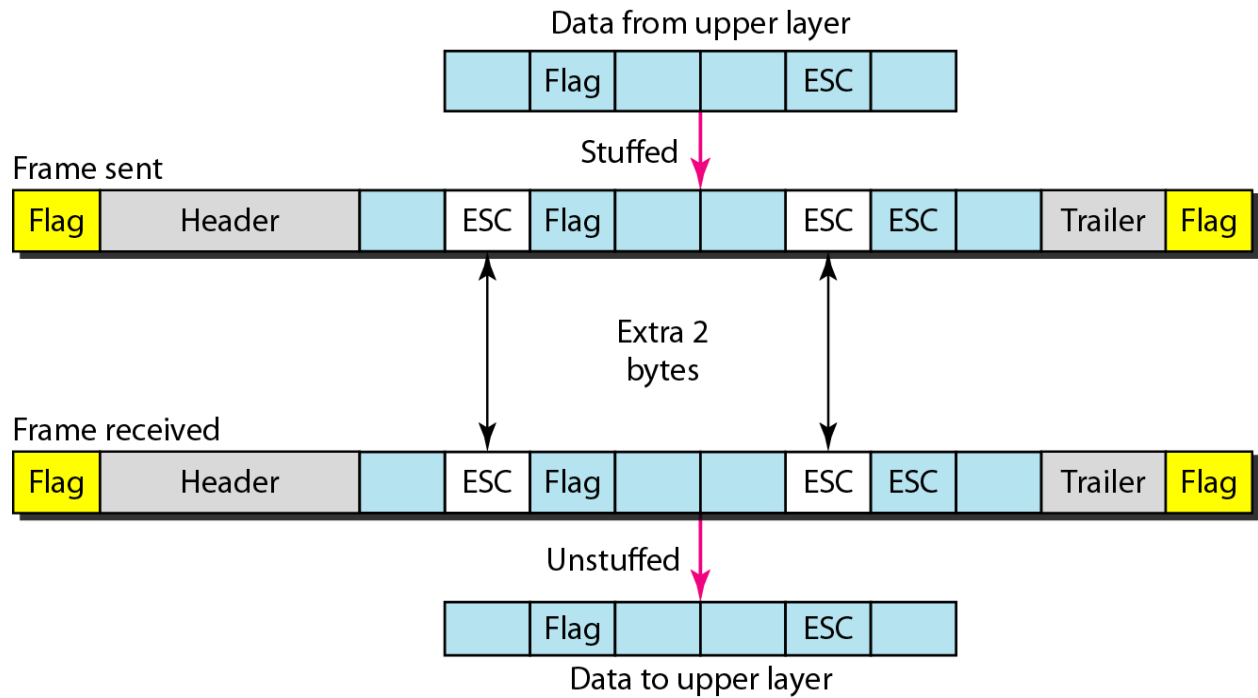
© Encyclopædia Britannica, Inc.

## Module-3

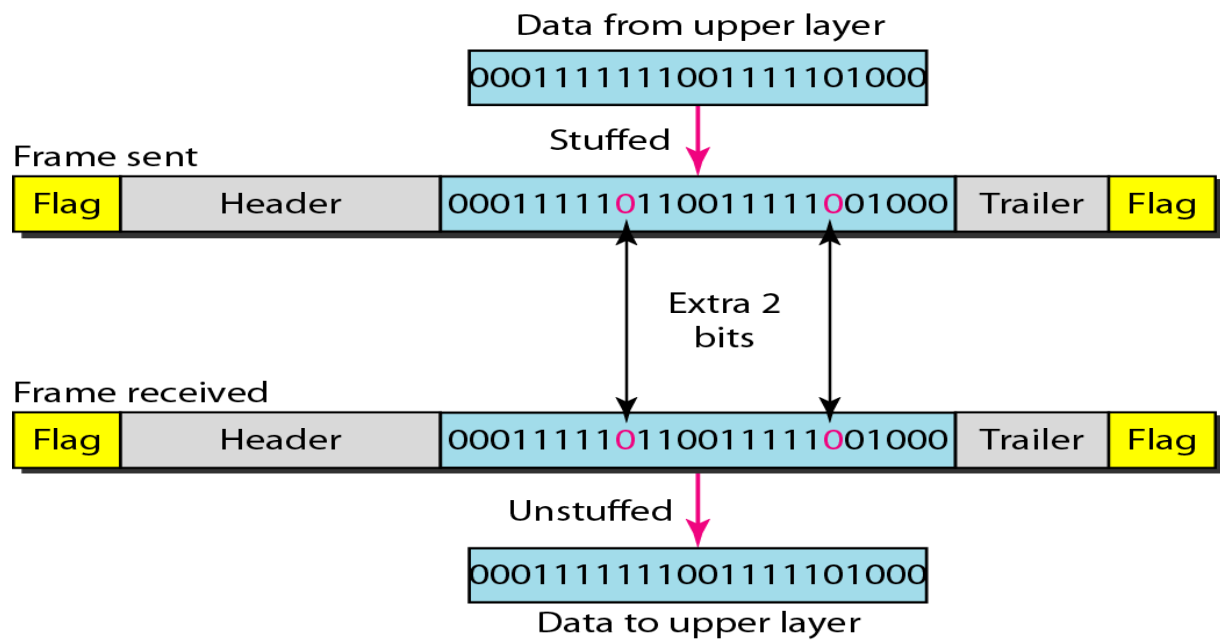
### *A frame in a character-oriented protocol*



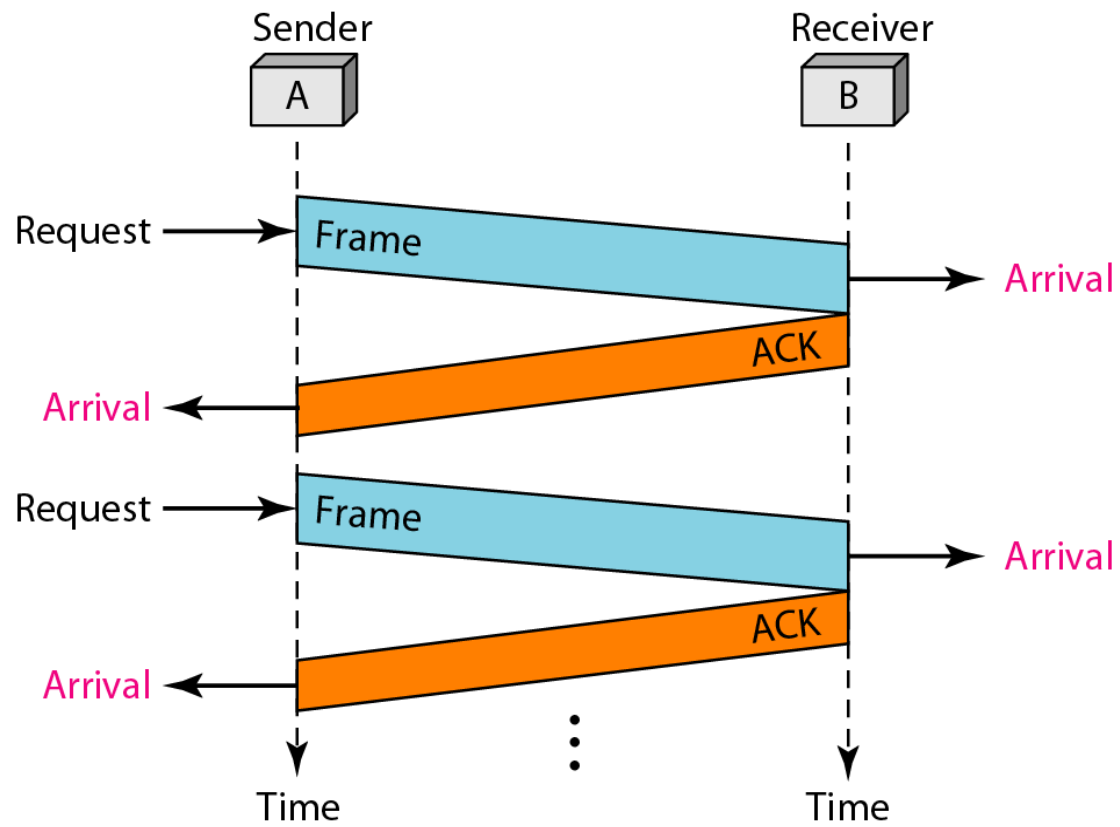
## Byte stuffing



## Bit stuffing and unstuffing



# Stop-and-Wait Protocol



The diagram illustrates the Stop-and-Wait protocol between a Sender (A) and a Receiver (B). A vertical line on the left represents the progression of time, starting with a green dot labeled 'Start timer' and ending with a red dot labeled 'Stop timer'.

**Sender (A):** Represented by a box at the top. Below it, a sequence of frames is shown as horizontal bars with segments numbered 0 to 7. The segments from 0 to 6 are labeled  $S_f$  and the segment from 7 to 12 is labeled  $S_n$ . The frames are:
 

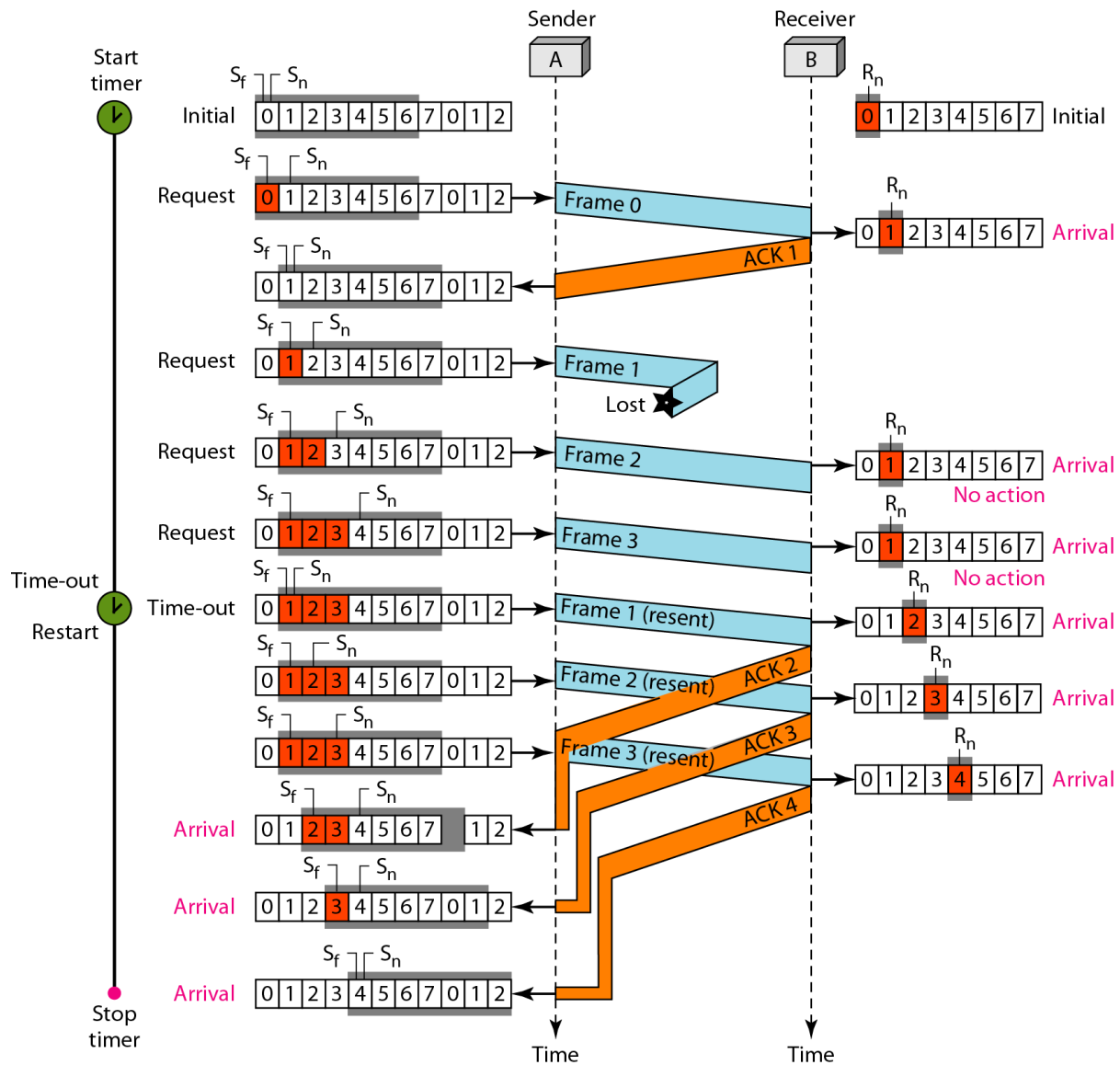
- Initial:** All segments 0-12 are white.
- Request:** Segment 0 is red.
- Arrival:** Segment 0 is white, segment 1 is red.
- Request:** Segment 1 is red.
- Request:** Segments 1 and 2 are red.
- Request:** Segments 1, 2, and 3 are red.
- Arrival:** Segment 3 is red.
- Arrival:** Segment 4 is red.

**Receiver (B):** Represented by a box at the top. Below it, a sequence of frames is shown as horizontal bars with segments numbered 0 to 7. The segments from 0 to 6 are labeled  $R_n$  and the segment from 7 to 12 is labeled  $R_s$ . The frames are:
 

- Initial:** All segments 0-12 are white.
- Arrival:** Segment 1 is red.
- Arrival:** Segment 2 is red.
- Arrival:** Segment 3 is red.
- Arrival:** Segment 4 is red.

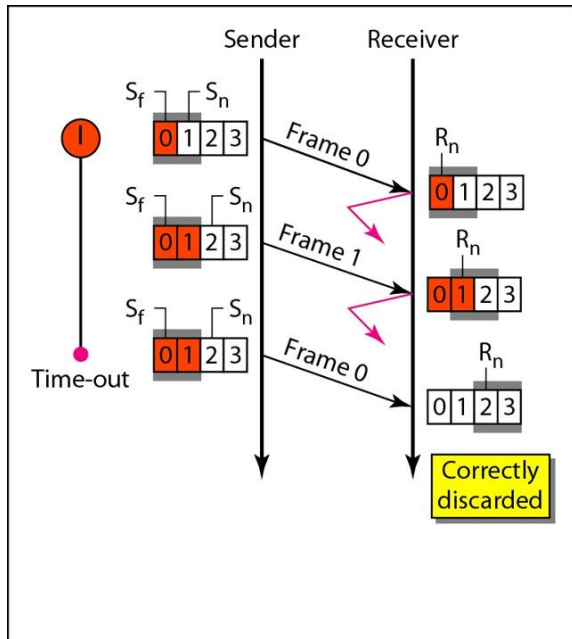
**Communication:**

- Frame 0:** A blue arrow points from Sender A to Receiver B.
- ACK 1:** An orange arrow points from Receiver B to Sender A.
- Frame 1:** A blue arrow points from Sender A to Receiver B.
- ACK 2:** An orange arrow points from Receiver B to Sender A.
- Frame 2:** A blue arrow points from Sender A to Receiver B.
- Lost:** A black star with the word 'Lost' indicates that Frame 2 was not received.
- ACK 3:** An orange arrow points from Receiver B to Sender A.
- Frame 3:** A blue arrow points from Sender A to Receiver B.
- ACK 4:** An orange arrow points from Receiver B to Sender A.

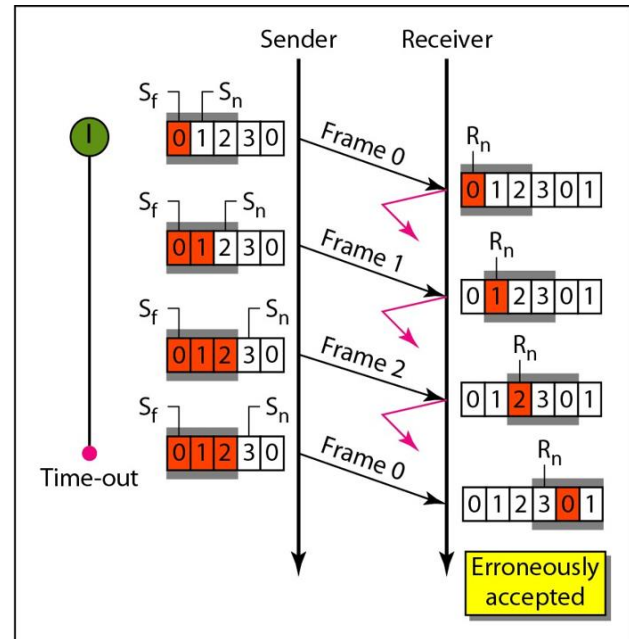




# Selective Repeat ARQ

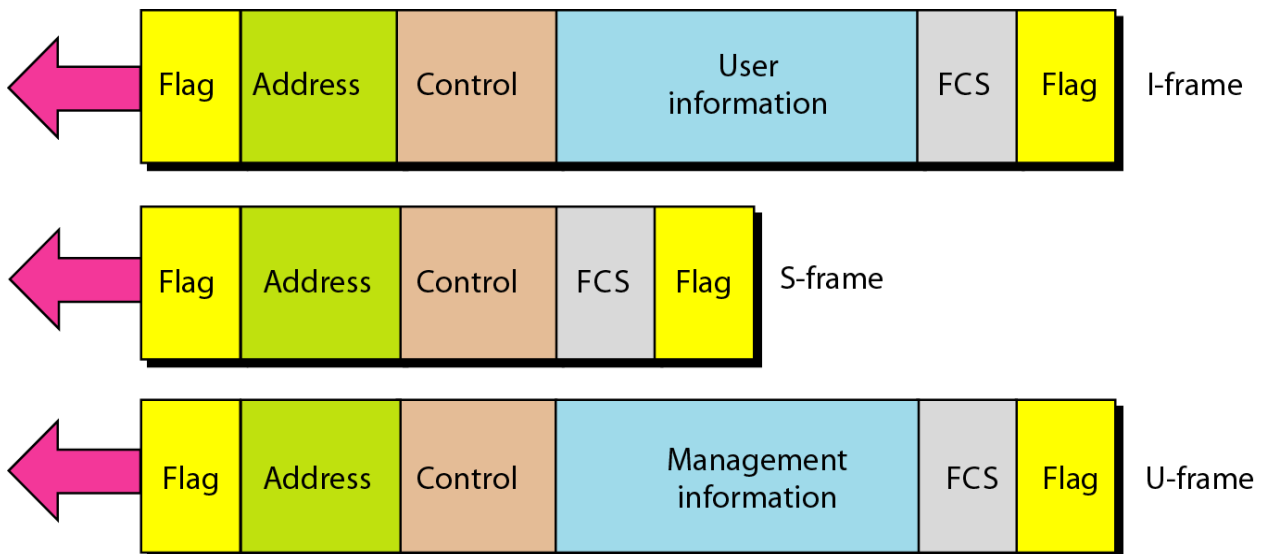


a. Window size =  $2^m - 1$

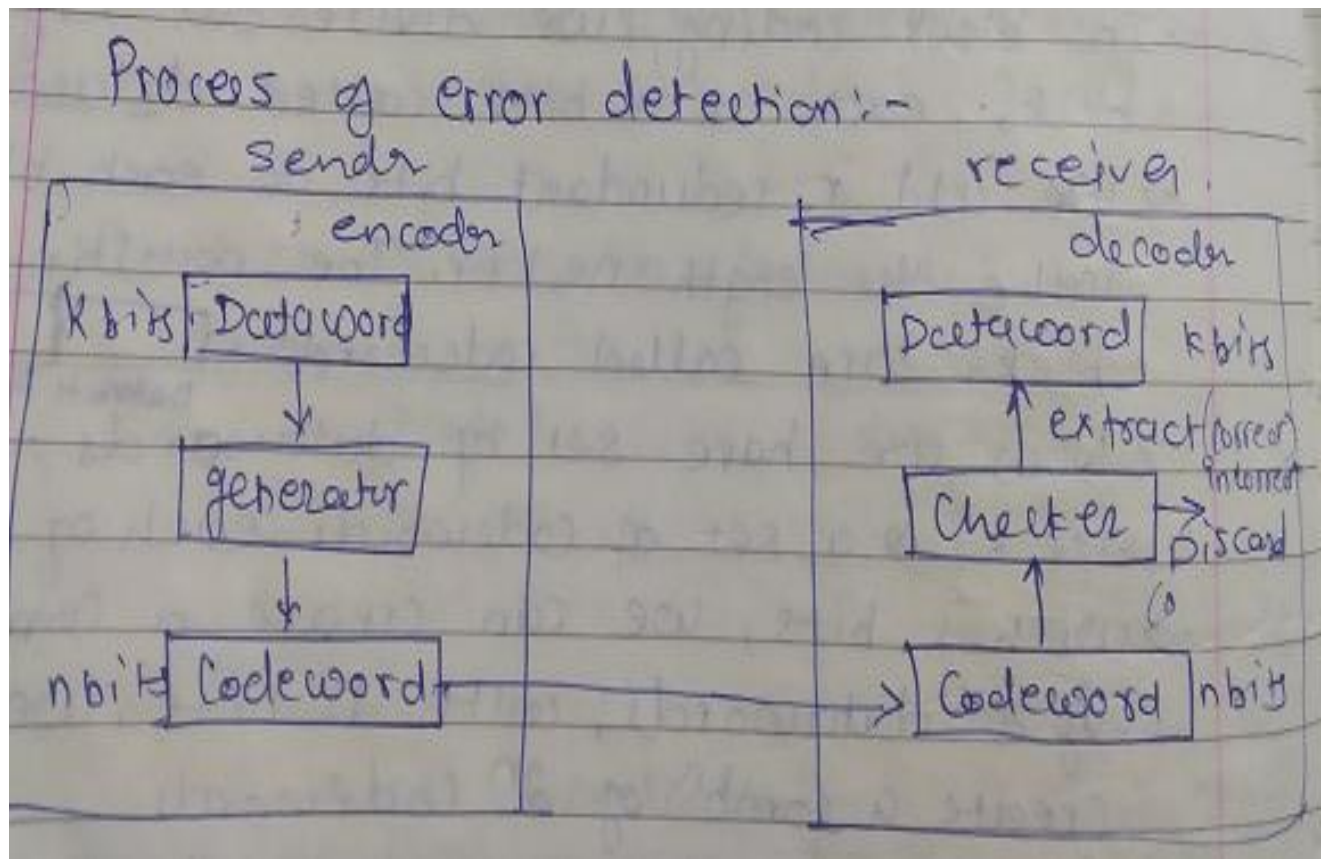


b. Window size >  $2^m - 1$

## HDLC frames



## Working of error detection



# Checksum

Original Data

10011001	11100010	00100100	10000100
----------	----------	----------	----------

1

2

3

k=4, m=8

Reciever

Sender

```

1  10011001
2  11100010
   -----
   101111011
   1
   -----
   01111100
3  00100100
   -----
   10100000
   10000100
   -----
   100100100
   1
   -----

```

Sum: 00100101

Checksum: 11011010

```

1  10011001
2  11100010
   -----
   101111011
   1
   -----
   01111100
3  00100100
   -----
   10100000
   10000100
   -----
   100100100
   1
   -----
   00100101
   11011010
   -----

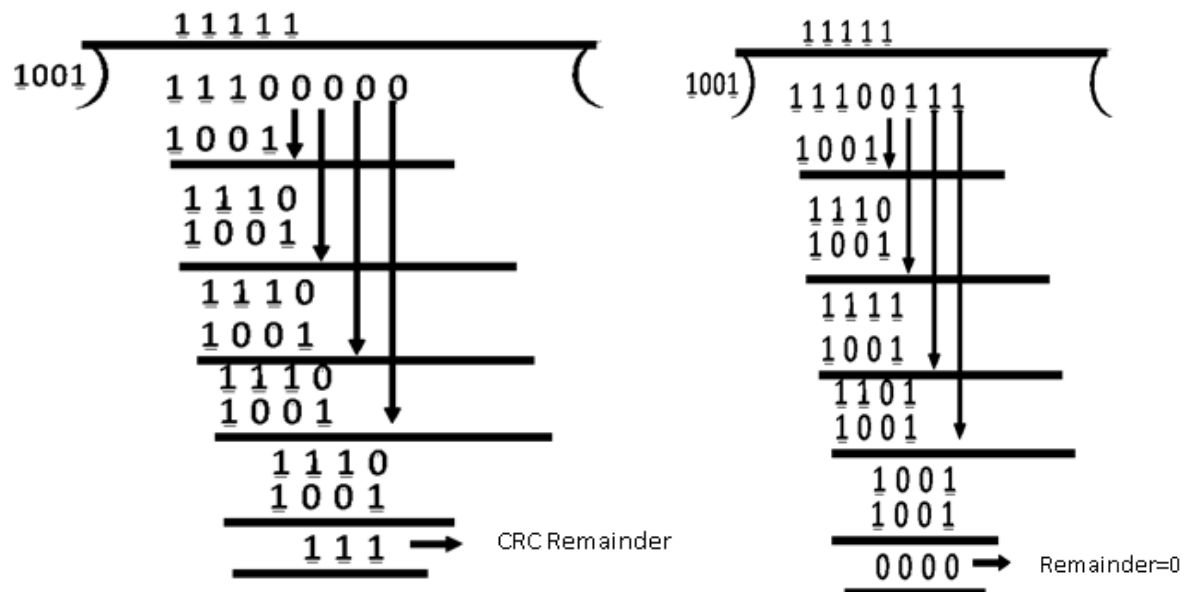
```

Sum: 11111111

Complement: 00000000

Conclusion: Accept Data

# CRC



## Hamming code

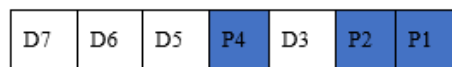


Fig. 5.22 a) 7-bit hamming code  $C(7,4)$



Fig. 5.22 b) 15-bit hamming code  $C(15,11)$

**Step 1: The codeword format:**

Given data bit = 1011

D7=1	D6=0	D5=1	P4	D3=1	P2	P1
------	------	------	----	------	----	----

P4, P2 and P1 is to be decided.

**Step 2: Decide P1:**

For P1, sections to be considered are 1,3,5,7

Here, we have to set P1=1 as 3,5,7=111 in order to have the even parity.

D7=1	D6=0	D5=1	P4	D3=1	P2	P1=1
------	------	------	----	------	----	------

**Step 3: Decide P2**

For P2, sections to be considered are 2,3,6,7

Here, we have to set P2=0 as 3,6,7=101 in order to have the even parity.

D7=1	D6=0	D5=1	P4=0	D3=1	P2=0	P1=1
------	------	------	------	------	------	------

Thus, the code word which is transmitted to the receiver = 1010101

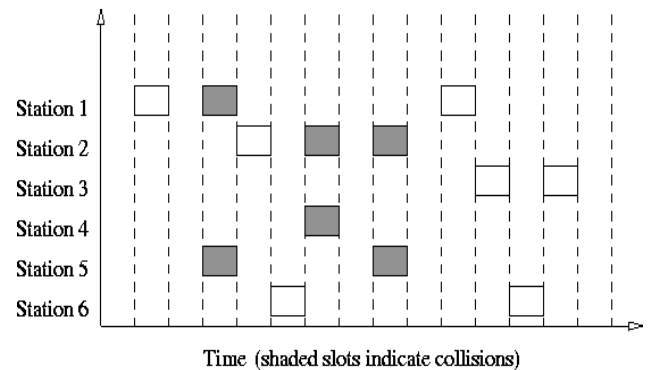
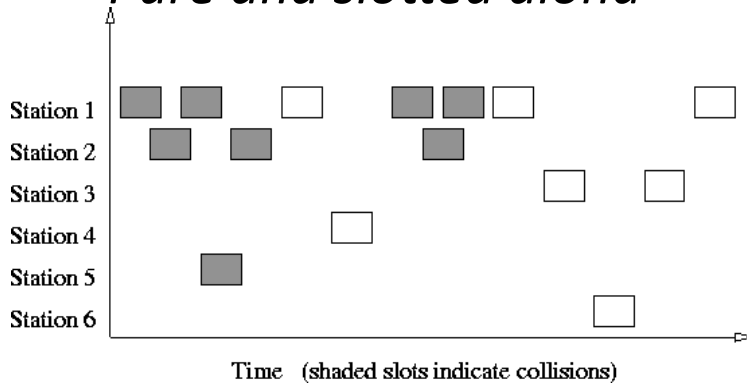
D7=1	D6=0	D5=1	P4	D3=1	P2=0	P1=1
------	------	------	----	------	------	------

**Step 4: Decide P4**

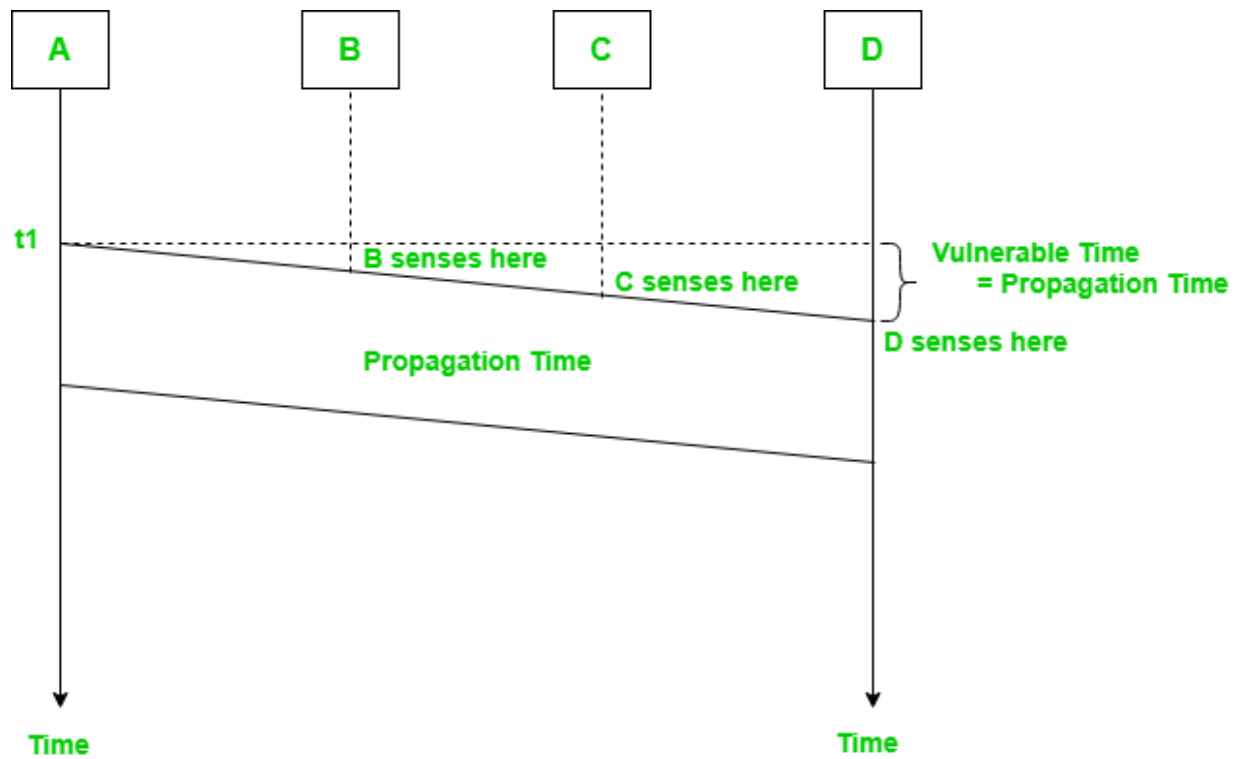
For P4, sections to be considered are 4,5,6,7

Here, we have to set P4=0 as 5,6,7=101 in order to have the even parity.

## Pure and slotted aloha

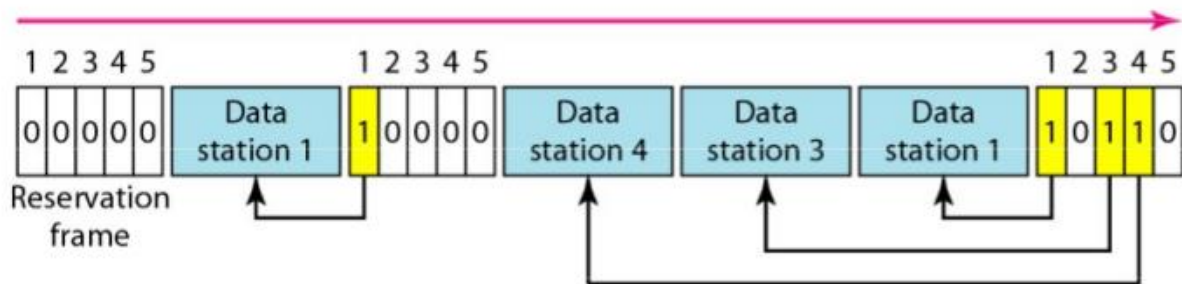


## Csma

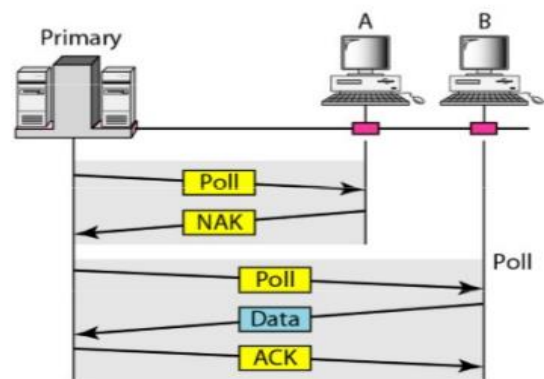
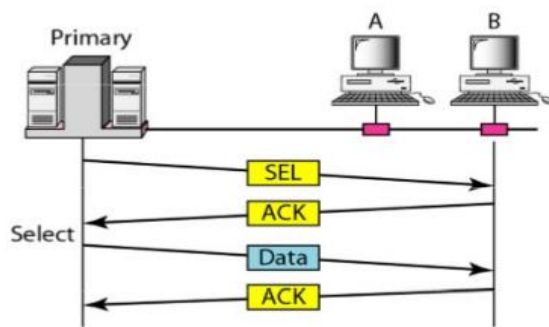


## Controlled access

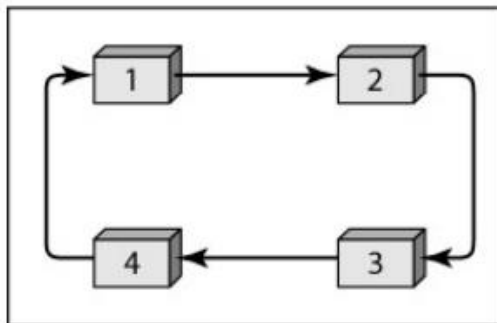
### Reservation



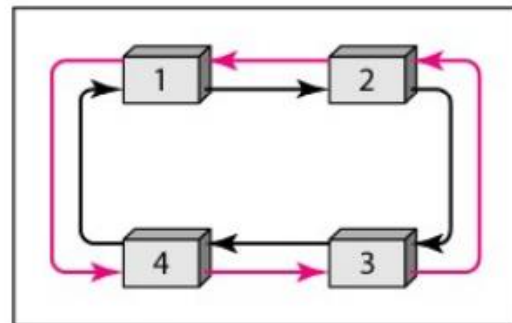
### Polling



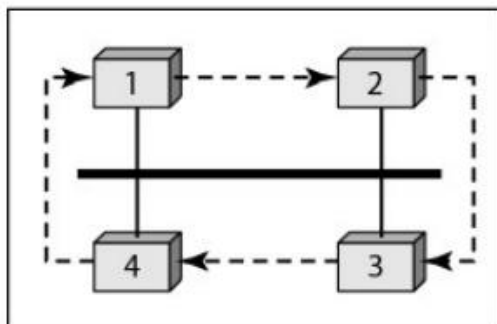
## Token passing



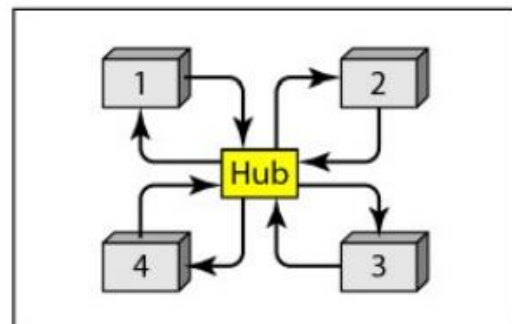
a. Physical ring



b. Dual ring



c. Bus ring

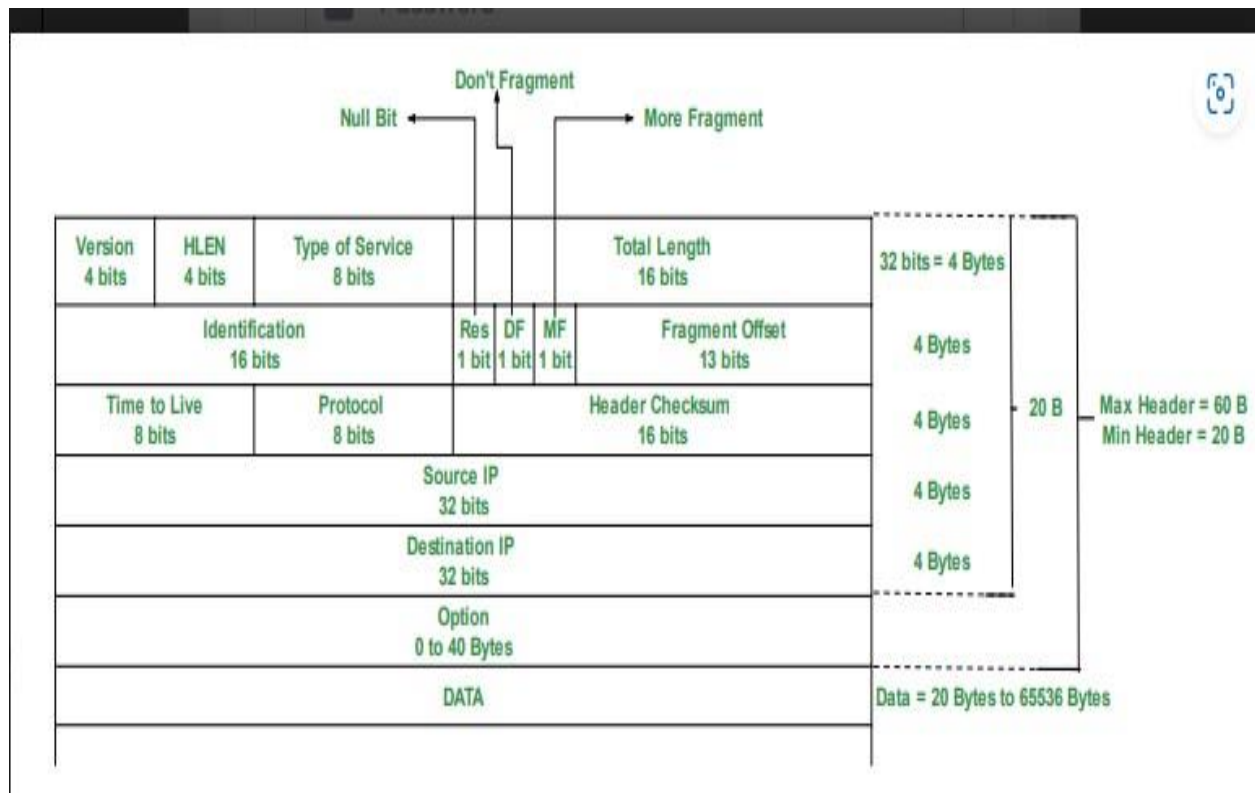


d. Star ring

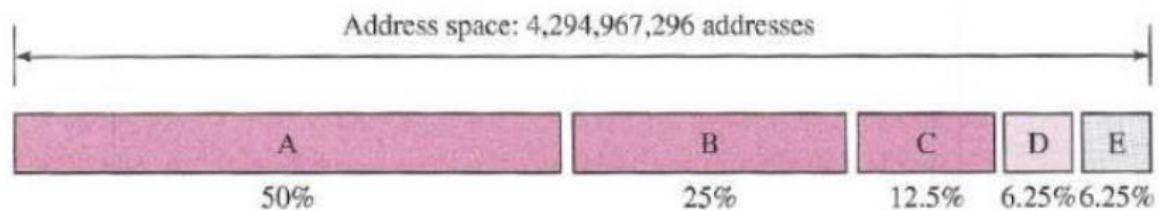
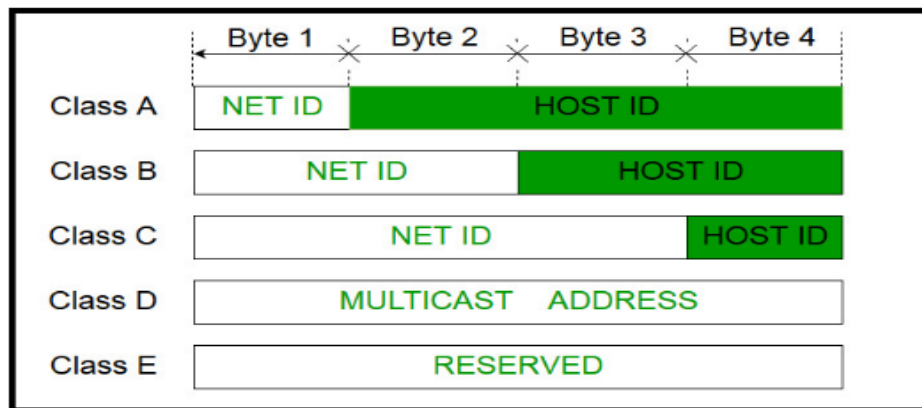
## Module-4

0	4	8	16	19	24	31
Version	IHL	Type of Service		Total Length		
Identification				Flags	Fragment Offset	
TTL		Protocol		Header Checksum		
Source IP Address						
Destination IP Address						
Options					Padding	

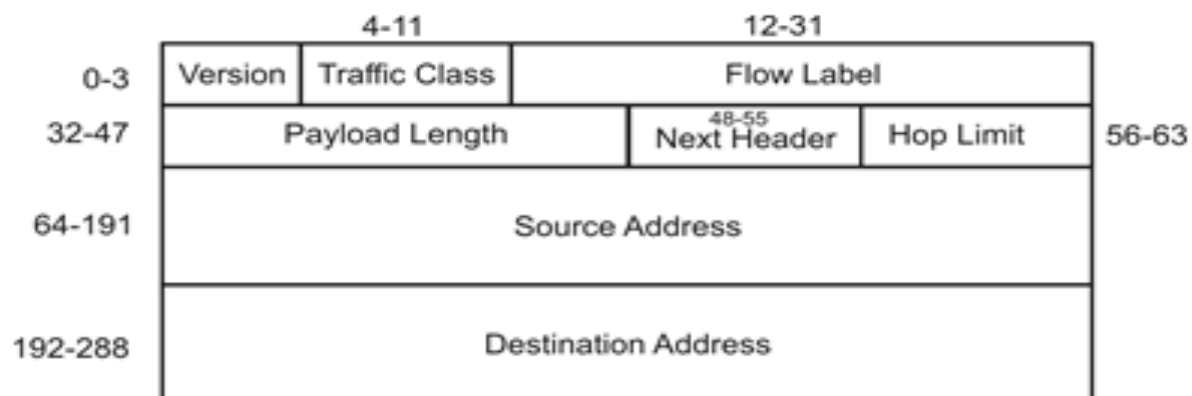
Fig: IPv4 Frame Format







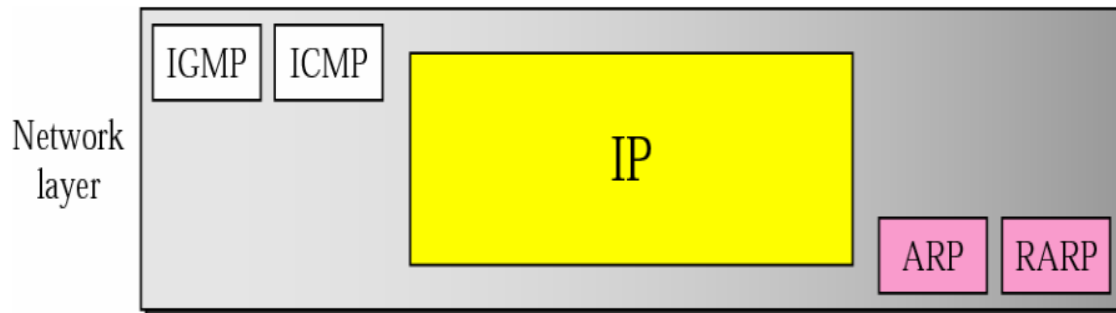
	8 bits	8 bits	8 bits	8 bits		
Class A	0	Prefix	Suffix		Class	Prefixes
Class B	10	Prefix	Suffix			First byte
Class C	110	Prefix	Suffix		A	$n = 8$ bits
Class D	1110	Multicast addresses			B	$n = 16$ bits
Class E	1111	Reserved for future use			C	$n = 24$ bits
					D	Not applicable
					E	Not applicable



**Fig 4.20.1: IPv6 Fixed Header**

# *Position of ARP and RARP in TCP/IP Protocol Suite*

---



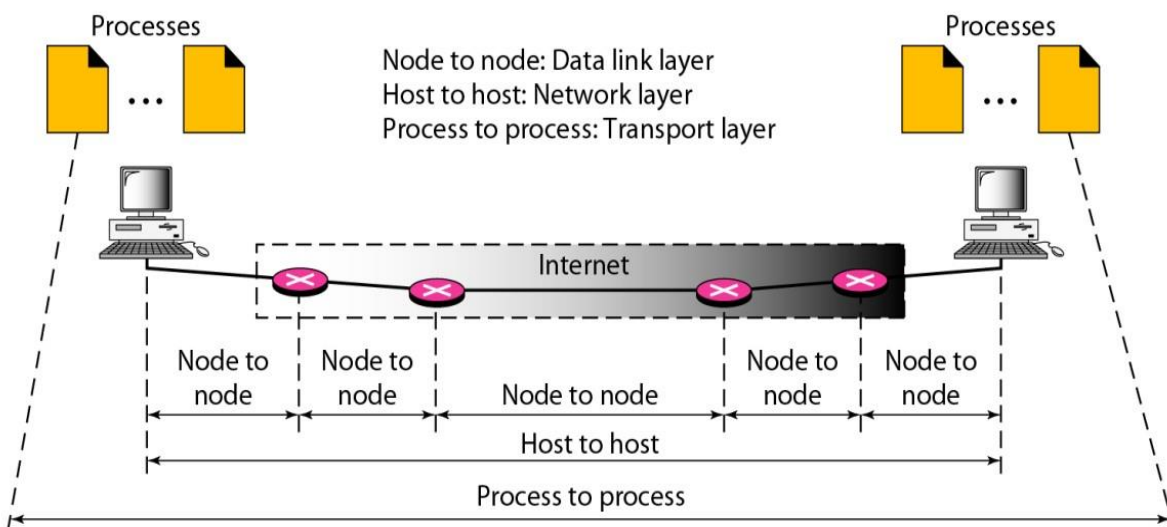
## ***ARP Packet***

Hardware Type		Protocol Type
Hardware length	Protocol length	<b>Operation</b> Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

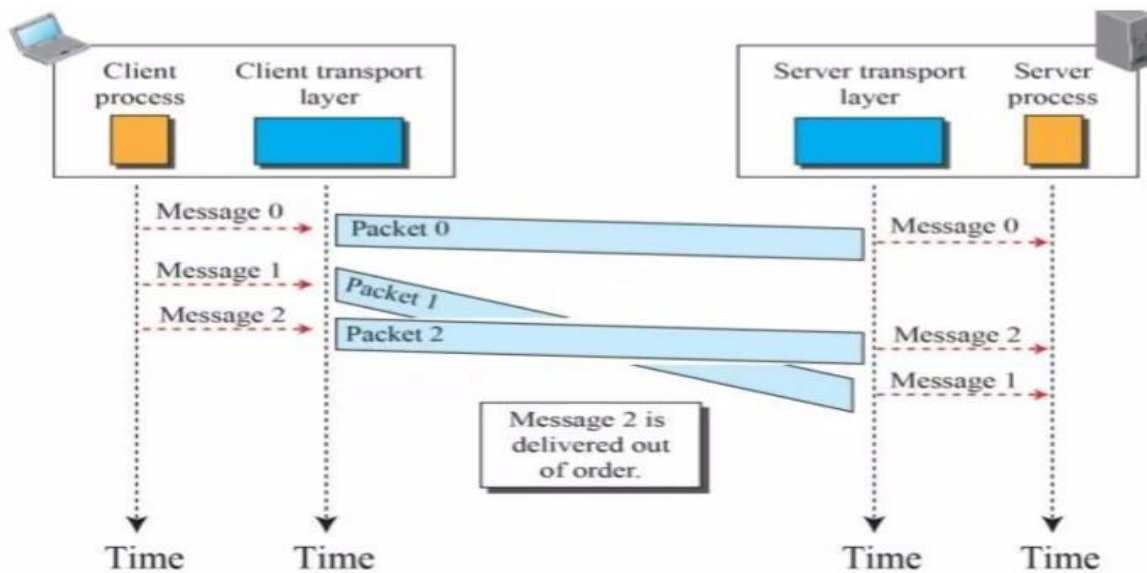
## ***RARP packet***

Hardware type		Protocol type
Hardware length	Protocol length	Operation Request 3, Reply 4
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP) (It is not filled for request)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled for request)		
Target protocol address (For example, 4 bytes for IP) (It is not filled for request)		

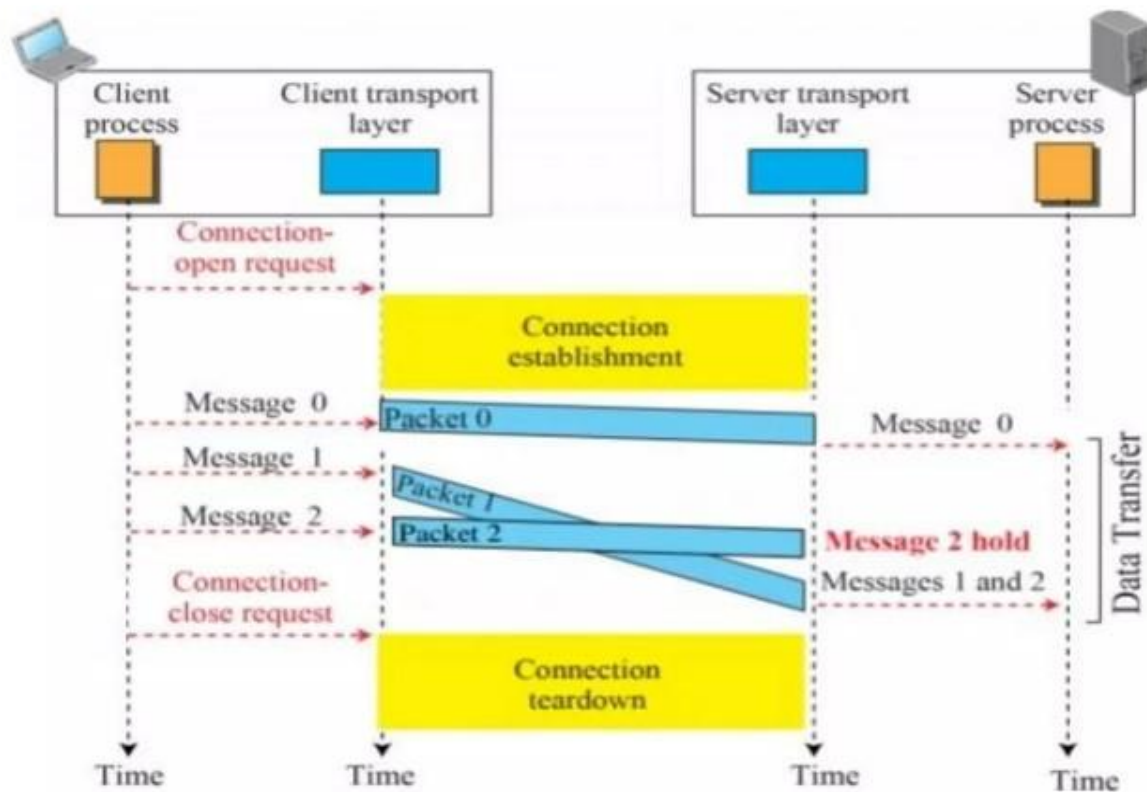
## Module-5

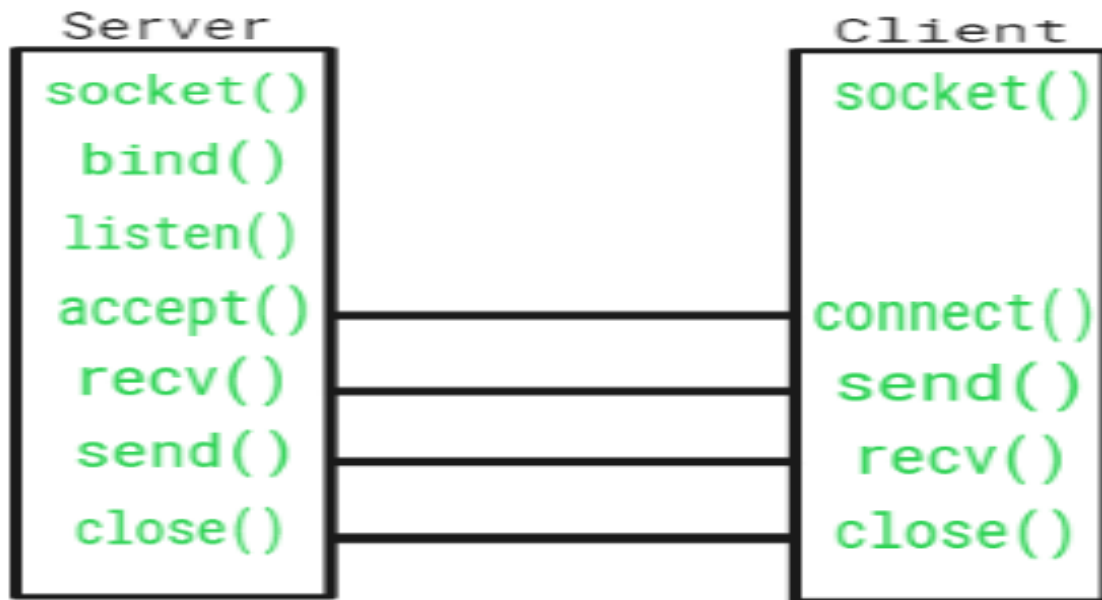


## Connectionless

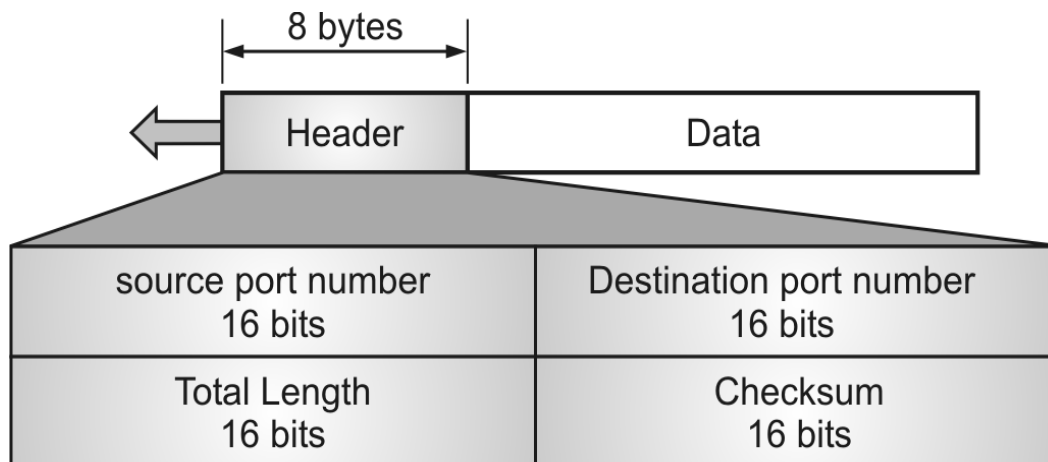


## Connection oriented

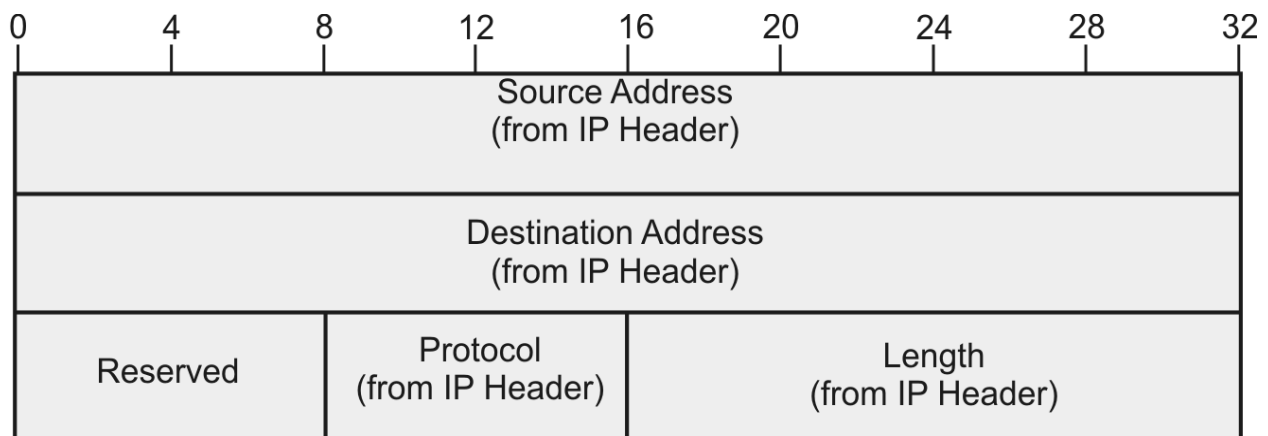




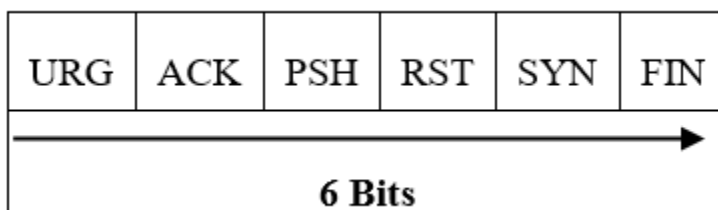
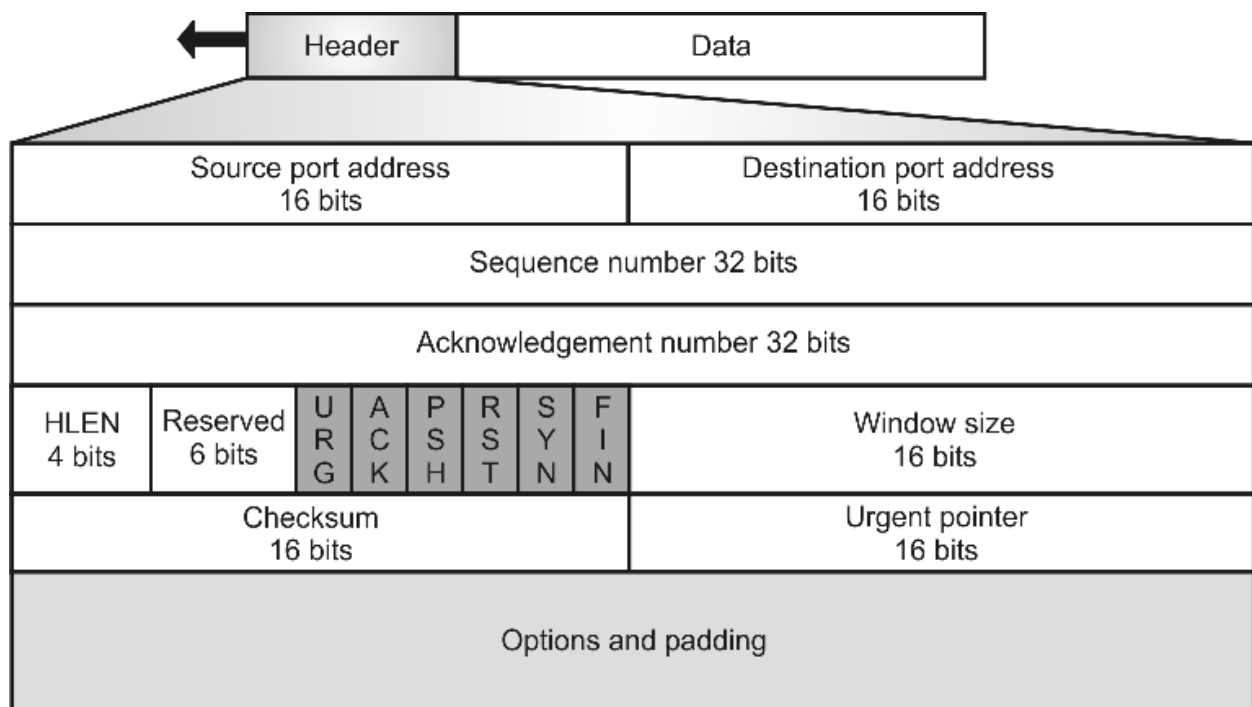
## UDP header



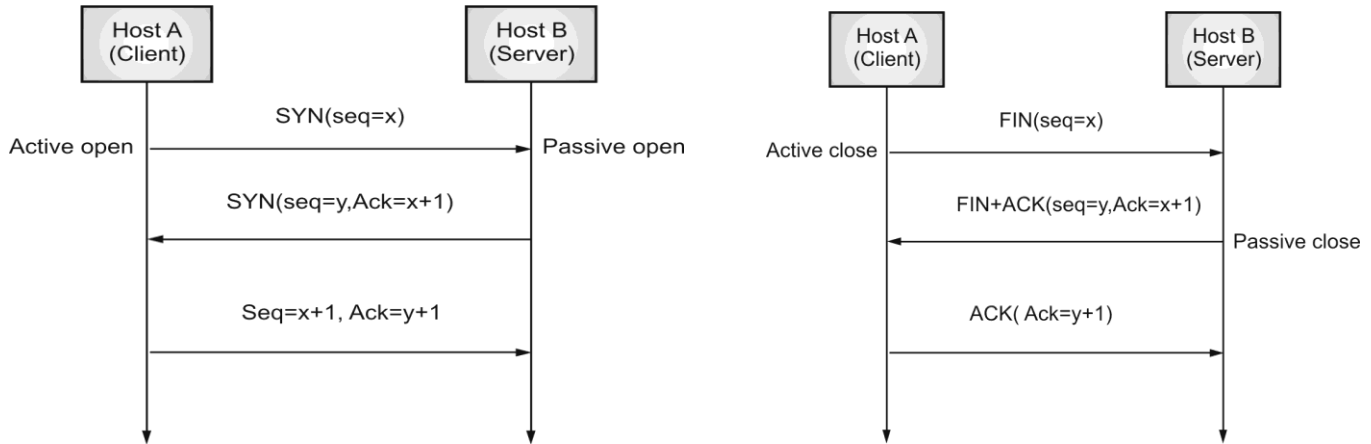
## UDP pseudo header



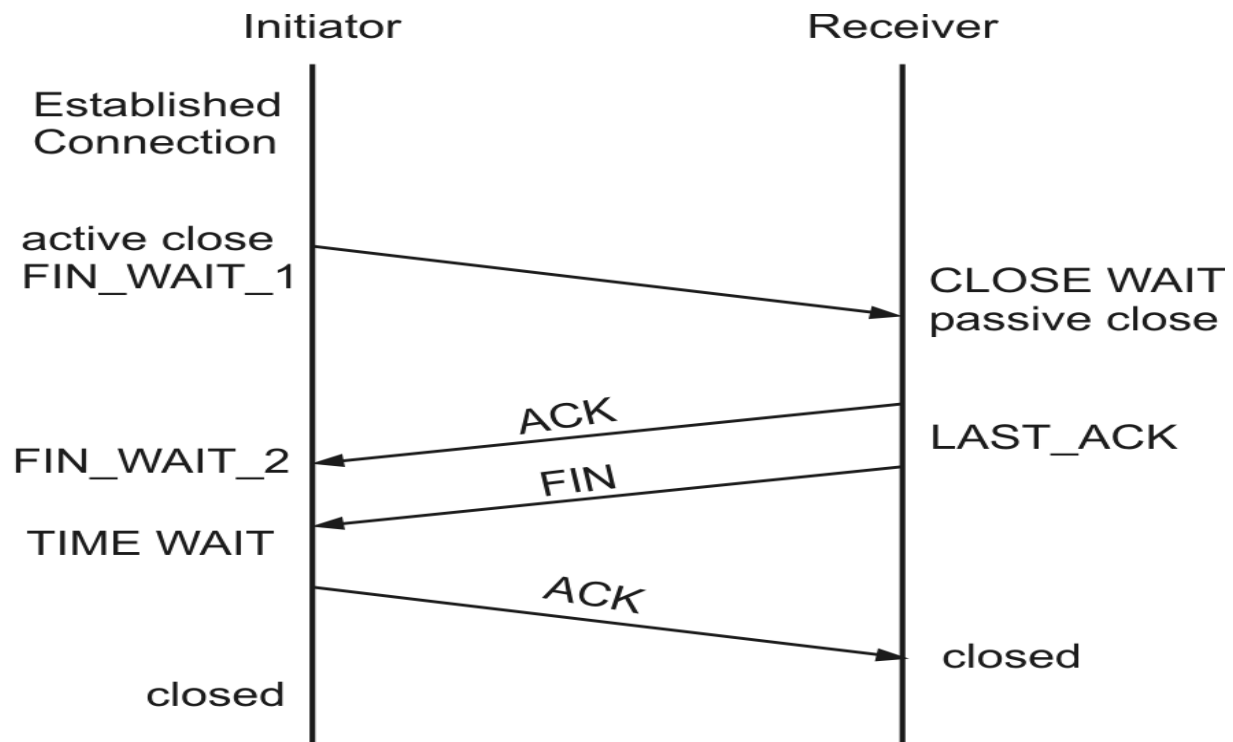
## TCP segment header



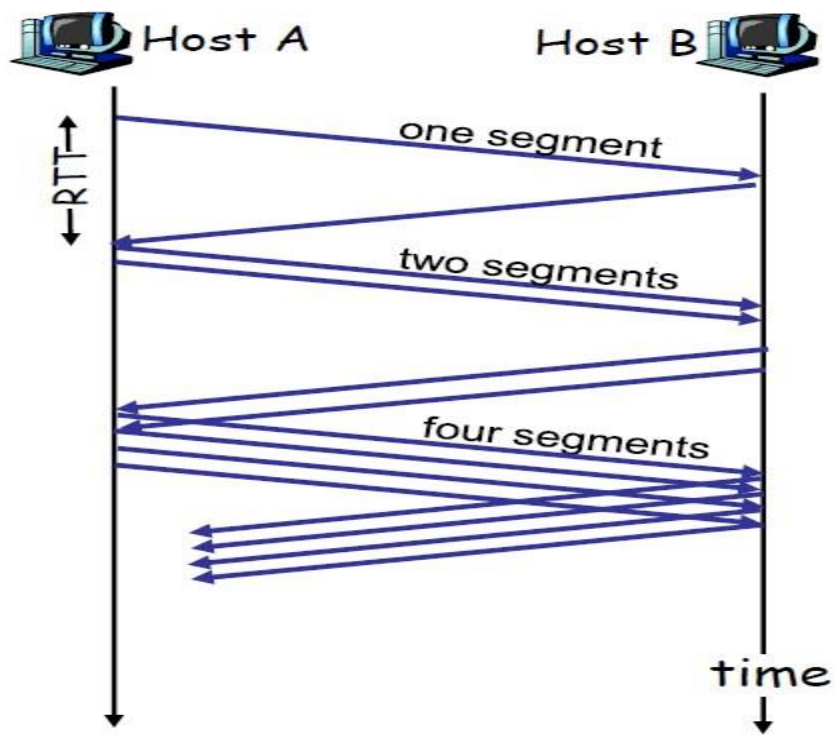
# Three way handshake



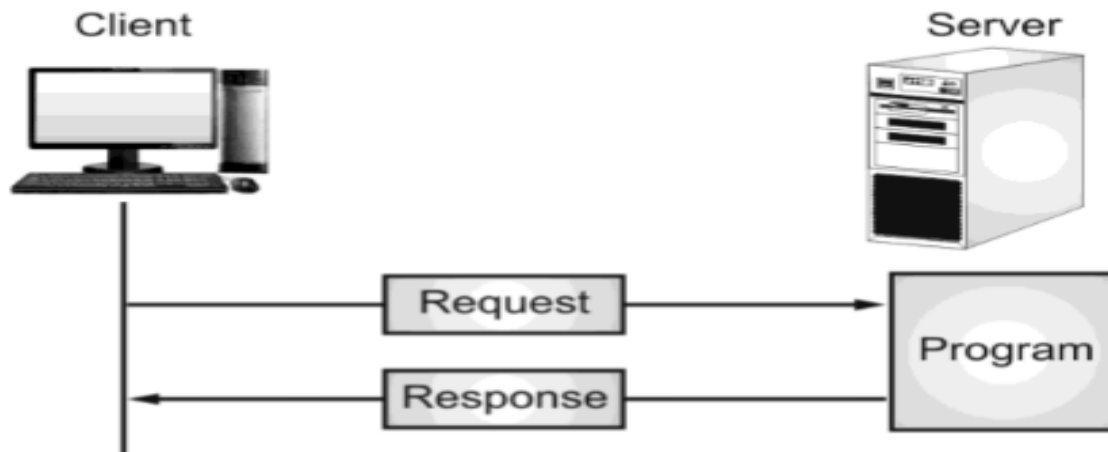
## 4 way handshake



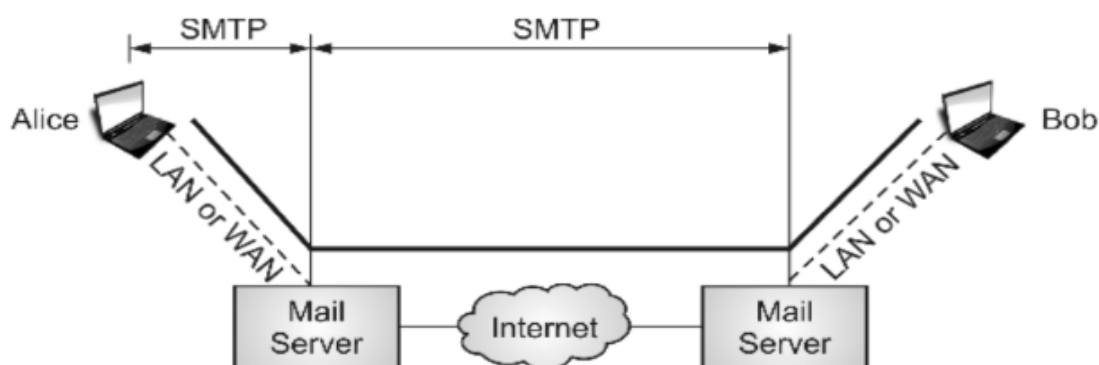
## ***TCP slow start***



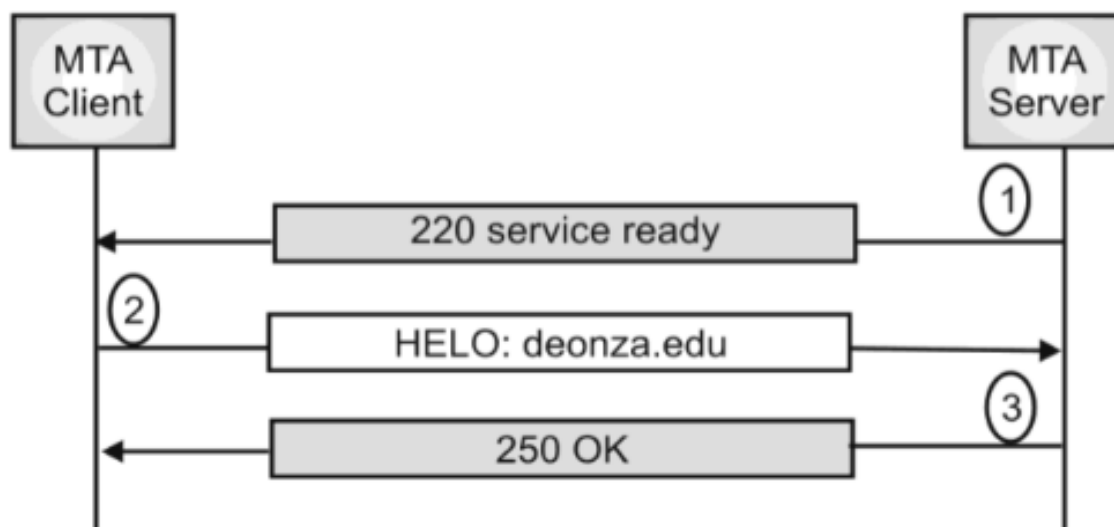




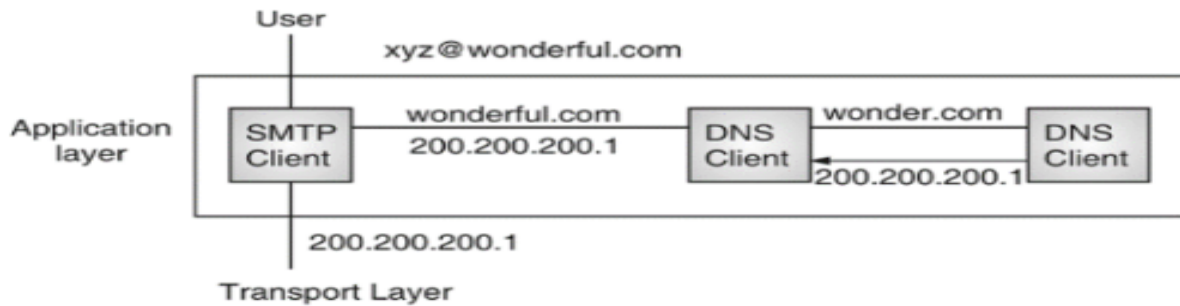
**Fig. 5.1.9 : HTTP Transaction**



**Fig. 5.3.7 : Range of SMTP Protocol**



**Fig. 5.3.13 : Connection Establishment in SMTP**



SMTP - Simple Mail Transfer Protocol (e-mail)  
 SMTP - Domain Name System

**Fig. 5.4.1 : Example of DNS**

## Module-6

# Bluetooth Protocol Stack

